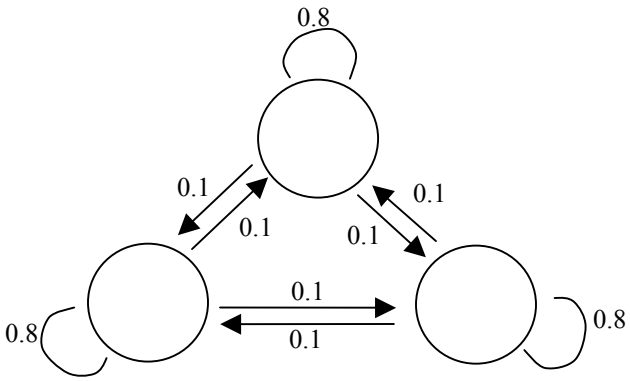
BioE/MCB/PMB C146/C246: Topics in Computational Biology and Genomics

Lecture 18) Hidden Markov Models

3 April 2003

Scribe: Susan Lin *Reader*:



Series of Observations
Can be used to estimate the emission probability of certain states.
Have a set of observations where the hidden values and observes are,
Taken a set of good examples of genes, HMMs can be built and used to search for others, i.e. search transembrane proteins by searching for hydrophobic stretches

Example: Meteorological grad student captured and driven for two hours. Using what is known about Berkeley weather to estimate weather of current location.

SSSFFRRRRSSFRRRR
UNUUNUNUNUUNUUNN

	S	F	R
S	III	II	
F		I	III
R	I		IIIIII

Prior knowledge of parameters is usually known.
Use Viterbi algorithm to calculate most probable path through data. This can be used to retrain the model. Don't know that it is correct but can assume so; therefore the parameters can be re-estimated by counting transitions and emissions

Viterbi Training

Using a gap or some prior knowledge, applying Viterbi to get most probable path and than assuming the most probable path is the training data, and using this data to calculate transition parameters, must use pseudocounts with small amounts of data.
Transition counts:

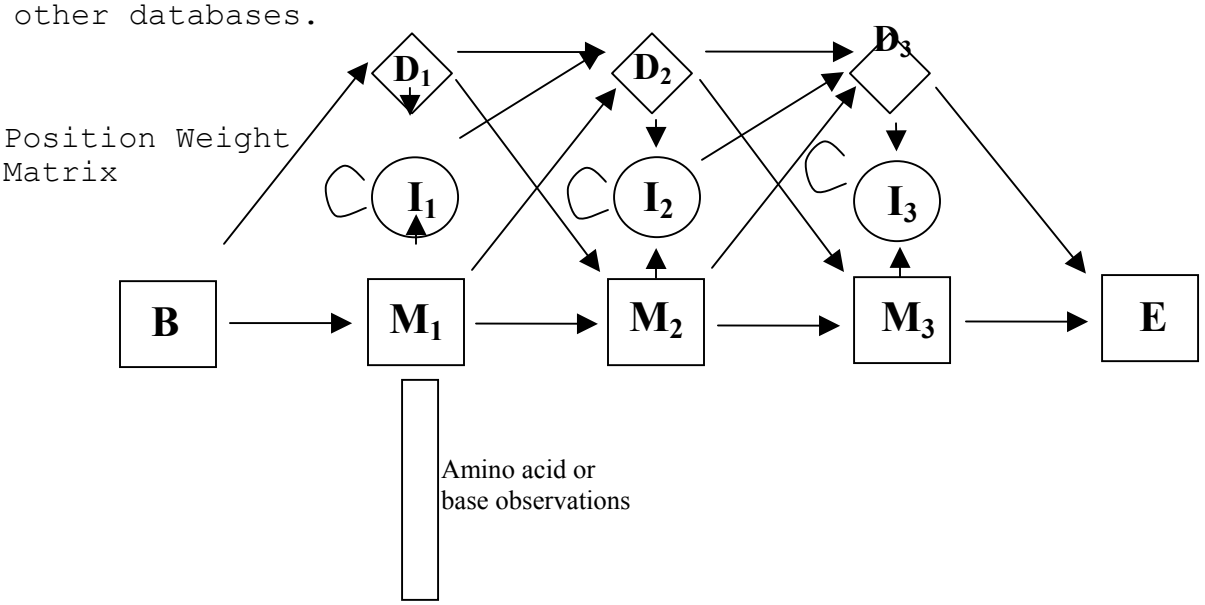
A_{SS} = 0.6 A_{FS} = 0 A_{RS} = 0.14
A_{SF} = 0.4 A_{FF} = 0.3 A_{RF} = 0
A_{SR} = 0 A_{FR} = 0.6 A_{RR} = 0.86

Pseudocounts – to avoid zeros, add .1 to the entire column

Training set is never perfect

Example: gene vs. non-gene, need a good idea of where genes aren't
Motif detection – a solid motif is hard to determine since there is a lot of noise.
The problem with HMM is to come up with the right data to train a model on.
The strength of a HMM is that you have a discrete Markovian model behind it and a discrete set of data.

Profile HMMs that lie behind PFAM
PFAM – database of protein domain families.
- The goal is to take a highly curated protein collection and align protein to build a HMM model.
- An annotation tool to take features of a domain and be able to apply it to other databases.



Example of paths through the model:

VGA--HAGEY	B→M_1→M_2→M_3→M_4→M_5→M_6→M_7→M_8→E	Observations:
V----NVDEV	B→M_1→D_2→D_3→M_4→M_5→M_6→M_7→M_8→E	Pseudocounts
VEA--DVAGH	B→M_1→M_2→M_3→M_4→M_5→M_6→M_7→M_8→E	M_1M_2 : 6 times → 0.7
VKG------D	B→M_1→M_2→M_3→D_4→D_5→D_6→D_7→M_8→E	M_1D_2 : 1 time → 0.2
VYS--TYETS	B→M_1→M_2→M_3→M_4→M_5→M_6→M_7→M_8→E	M_1I_1 : 0 times → 0.1
FNA--NIPKH	B→M_1→M_2→M_3→M_4→M_5→M_6→M_7→M_8→E	
IAGADNGAGV	B→M_1→M_2→M_3→I_3→I_3→M_4→M_5→M_6→M_7→M_8→E	

How do you construct parameters of model?
Not like a normal matrix, instead, figure out the path through each of these states.

To determine whether a domain is contained within the sequences apply the **Viterbi algorithm** to the sequence.

Score: Log (P(x, Profile HMM) / P(x | B)) for x_1, x_2, … x_n
V_j^M (i) for example: V_j^{M4}(L) "i" = emitted

What is the probability of x given our profile HMM compared to the probability of x in the background model?

- come up with parameters: at M position j, the j matrix position, what is the log odds ratio probability of submodel up to that point, log

odds ratio or equivalent to the probability of the path that ends at the n state at position 4 emitting character x

- probability of the most probable path that ends at the end state at position 4 submitting character L, end up looking at the score in relation to other sequences

take the sequence and calculate the most probable alignment through the path which will indicate where the domains will be, while evaluating the prob of the path based on the HMM compared to the probability of the background model.

Recursion equations are the same as the viterbi except the path to a state is different:

There are 3 ways to get to a certain position V_3M_L

- from M_2, I_2, D_2

for I state, the path can come from a match state, delete state, or itself, an insert state

Algorithm: Virterbi (page 56, Durbin)

Initialization (i=0): $v_o(0) = 1$, $v_k(0) = 0$ for $k > 0$

Recursion (i=1…L): $v_i(i) = e_i(x_i) \max_k (v_k (i-1) a_{kl})$

$Ptr_i(l) = \text{argmax}_k(v_k(i-1) a_{kl})$

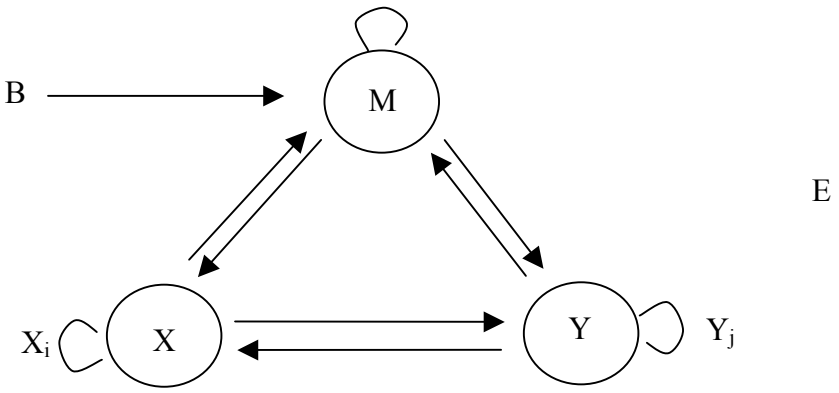
Termination: $P(x,\Pi^*) = \max_k (v_k (L) a_{k0})$;

$\Pi^*_L = \text{argmax}_k (v_k (L) a_{k0})$

Traceback (i=L…1): $\Pi^*_{i-1} = ptr_i(\Pi^*_i)$

Production of alignment from HMM:

Generating an alignment in which there are always 2 characters coming up next to one another



- emits aligned bases
- or emits a base for one or the other sequence

This can be used to calculate the probability 2 sequences are aligned over all other possible alignments.

You can show that different parameterizations and structures correspond to different gap penalties and affine gap penalties.

You can do all the same kind of alignments with SW with this HMM but can include:

- the probability the two sequences are aligned summed over all possible alignments, using forward and backward algorithm

- calculate the posterior probability of different events, using forward and backward algorithm; useful for assessing which parts of the algorithm are good or bad compared to all possible alignments
- calculation of the probability that a given base is aligned to another base